Diffusion Map Kernel Analysis for Target Classification

Jason C. Isaacs, *Member, IEEE*

***Abstract*— Given a high dimensional dataset, one would like to be able to represent this data using fewer parameters while preserving relevant information, previously this was done with principal component analysis, factor analysis, or feature selection. However, if we assume the original data actually exists on a lower dimensional manifold embedded in a high dimensional feature space, then recently popularized approaches based in graph-theory and differential geometry allow us to learn the underlying manifold that generates the data. One such manifold-learning technique, called Diffusion Maps, is said to preserve the local proximity between data points by first constructing a representation for the underlying manifold. This work examines binary target classification problems using Diffusion Maps to embed the data with various kernel representations for the diffusion parameter. Results demonstrate that specific kernels are well suited for Diffusion Map applications on some sonar feature sets and in general certain kernels outperform the standard Gaussian and Polynomial kernels, on several of the higher dimensional data sets including the sonar data contrasting with their performance on the lower-dimensional publically available data sets.**

I. Introduction

THE central problem in high-dimensional data analysis is the trade-off between computational complexity and the resolution gained with either more features or pixels. Therefore, a typical first step in analyzing high-dimensional data is to find a lower-dimensional representation and the concise description of its underlying geometry and density. This is usually done however, with global dimension reducing techniques such as principal component analysis, and Multidimensional Scaling. These techniques in general work well with well behaved maximally variant data. What if the data is only locally correlated? Then these techniques do not provide informative embedded data. Alternatively, graph based manifold learning techniques offer to embed the data based on local relationship preservation, i.e., they generally preserve the neighborhood structure. Such techniques are Diffusion Maps [1] and [2], Local linear Embedding [3], Laplacian Eigenmaps [4], Hessian Eigenmaps [5], and Local Tangent Space Alignment[6].

In this paper we consider the manifold learning technique Diffusion Maps of Coifman et al. [1], [2] and analyze the neighborhood preserving effects of kernel selection on the resulting manifold for publicly available data sets. These effects are studied by looking at the classification results for each binary target data set in various embeddings.

Manuscript received August 21, 2009.

J. C. Isaacs is with the Naval Surface Warfare Center, Panama City, FL 32405. email: jason.c.isaacs1@navy.mil

II. Diffusion Maps

A. Overview

Diffusion Maps are defined as the embedding of complex data onto a low dimensional Euclidian space, via the eigenvectors of suitably normalized random walks over the given dataset. It has been shown, both theoretically in [1] and by examples in [2] how this embedding can be used for dimensionality reduction, manifold learning, geometric analysis of complex data sets and fast simulations of stochastic dynamical systems.

Diffusion Maps are said to preserve the local proximity between data points by first constructing a graph representation for the underlying manifold. The vertices, or nodes of this graph, represent the data points, and the edges connecting the vertices, represent the similarities between adjacent nodes. If properly normalized, these edge weights can be interpreted as transition probabilities for a random walk on the graph. After representing the graph with a matrix, the spectral properties of this matrix are used to embed the data points into a lower dimensional space, and gain insight into the geometry of the dataset. It has been shown in [1] and [2] that the eigenfunctions of Markov matrices can be used to construct coordinates called Diffusion Maps that generate these efficient representations of the complex geometric structures and the associated family of diffusion distances, obtained by iterating the Markov matrix, defines the multiscale geometries that prove to be useful in the context of data parameterization and dimensionality reduction. The process of constructing these Diffusion Maps as described in [1] and [2] is discussed in sections II.B through II.E.

B. Construction of a Random Walk on the Data

Given a data set Ω with a distribution μ of the points on Ω and a kernel $k : \Omega \times \Omega \rightarrow \mathrm{R}$ that satisfies the following properties:

- k is symmetric: $k(x, y) = k(y, x)$,
- k is positivity preserving: $k(x, y) \geq 0$.

This kernel represents some notion of affinity or similarity between points of Ω as it describes the relationship between pairs of points in this set and in this sense, one can think of the data points as being the nodes of a symmetric graph whose weight function is specified by k. The kernel constitutes an a priori presumption of the local geometry of Ω, and since a given kernel will capture a specific feature of the data set, its choice should be guided by the application

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUN 2010	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Diffusion Map Kernel Analysis for Target Classification	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **IEEE**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM202806. Proceedings of the Oceans 2009 MTS/IEEE Conference held in Biloxi, Mississippi on 26-29 October 2009. Copyright belongs to the Marine Technology Society., The original document contains color images.

14. ABSTRACT
Given a high dimensional dataset, one would like to be able to represent this data using fewer parameters while preserving relevant information, previously this was done with principal component analysis, factor analysis, or feature selection. However, if we assume the original data actually exists on a lower dimensional manifold embedded in a high dimensional feature space, then recently popularized approaches based in graph-theory and differential geometry allow us to learn the underlying manifold that generates the data. One such manifold-learning technique, called Diffusion Maps, is said to preserve the local proximity between data points by first constructing a representation for the underlying manifold. This work examines binary target classification problems using Diffusion Maps to embed the data with various kernel representations for the diffusion parameter. Results demonstrate that specific kernels are well suited for Diffusion Map applications on some sonar feature sets and in general certain kernels outperform the standard Gaussian and Polynomial kernels, on several of the higher dimensional data sets including the sonar data contrasting with their performance on the lower-dimensional publically available data sets.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**SAR**	**7**	

that one has in mind; this will be discussed later.

It is known that to any reversible Markov process, one can associate a symmetric graph. In addition, the converse is also true, i.e., from the graph defined by (Ω, *k*), one can construct a reversible Markov chain on Ω. This technique is known as the normalized graph Laplacian construction. The steps are as follows: define

$$d(x) = \int_{\Omega} k(x, y) d\mu(y) \tag{1}$$

to be a local measure of the degree of node *x* in this graph and define P^t to be an n × n matrix whose entries are given by

$$p_t(x, y) = \frac{k(x, y)}{d(x)} \tag{2}$$

which is the probability of transition from *x* to *y* in one time step. For *t* = 1 this can be interpreted as the first-order neighborhood structure of the graph.

C. Powers of P and Multiscale Geometric Analysis of Ω

The matrix P contains geometric information about the data set Ω. The transitions that it defines directly reflect the local geometry defined by the immediate neighbors of each node in the graph of the data. In other words, p_1(x, y) represents the probability of transition in one time step from node x to node y and it is proportional to the edge-weight $k(x, y)$. For $t \geq 0$, the probability of transition from *x* to *y* in *t* time steps is given by $p_t(x, y)$, the kernel of the t^{th} power P^t of P. Larger powers of P, allows the integration of the local geometry and therefore will reveal relevant geometric structures of Ω at different scales, i.e., larger neighborhoods.

D. Spectral Analysis of the Markov Chain

Powers of P constitute an object of interest for the study of the geometric structures of Ω at various scales. A classical way to describe the powers of an operator is to employ the language of spectral theory, namely eigenvectors and eigenvalues. Although for general transition matrices of Markov chains, the existence of a spectral theory is not guaranteed, the random walk constructed here exhibits very particular mathematical properties, i.e., if the graph is connected, which we now assume, then the stationary distribution is unique and we have

$$\lim_{t \to +\infty} p_t(x, y) = \phi_0(y) \tag{3}$$

where the Markov chain has a stationary distribution given by

$$\phi_0(y) = \frac{d(y)}{\sum_{z \in \Omega} d(z)}. \tag{4}$$

The chain is reversible, i.e., it follows the detailed balance condition:

$$\phi_0(x) p_1(x, y) = \phi_0(y) p_1(y, x). \tag{5}$$

The vector ϕ_0 is the top left eigenvector of P. The spectral analysis of the Markov chain is governed by the following eigen-decomposition

$$p_t(x, y) = \sum_{l \geq 0} \lambda_l^t \psi_l(x) \phi_l(y), \tag{6}$$

where $\{\lambda_l\}$ is the sequence of *eigenvalues* of *P* (with $|\lambda_0| \geq |\lambda_1| \geq |\lambda_2| \geq \cdots$) and $\{\psi_l\}$ and $\{\phi_l\}$ are the corresponding biorthogonal right and left eigenvectors.

E. Diffusion Distances and Diffusion Maps

The spectral properties of the Markov chain can now be linked to the geometry of the data set Ω. As previously mentioned, the idea of defining a random walk on the data set relies on the following principle: the kernel *k* specifies the local geometry of the data and captures some geometric feature of interest. The Markov chain defines fast and slow directions of propagation, based on the values taken by the kernel, and as one runs the walk forward, the local geometry information is being propagated and accumulated the same way local transitions of a system can be integrated in order to obtain a global characterization of this system.

Running the chain forward is equivalent to computing the powers of the operator P . For this computation, we could, in theory, use the eigenvectors and eigenvalues of P. Therefore, we are going to directly employ these objects in order to characterize the geometry of the data set Ω. The family of diffusion distances $\{D_t\}_{t \in N}$ is given by

$$D_t^2(x, z) = \sum_{y \in \Omega} \frac{\left(p_t(x, y) - p_t(z, y)\right)^2}{\phi_0(y)}. \tag{7}$$

In other words, $D_t(x, z)$ is a functional weighted l_2 distance between the two posterior distributions $p_t(x, \cdot)$ and $p_t(z, \cdot)$. For a fixed value of *t* , D_t defines a distance on the set Ω. By definition, the notion of proximity that it defines reflects the connectivity in the graph of the data. Indeed, $D_t(x, z)$ will be small if there is a large number of short paths connecting *x* and *z*, that is, if there is a large probability of transition from *x* to *z* and vice versa. The main interesting features of diffusion distance are: 1) the points are closer if they are highly connected, 2) $D_t(x, z)$ involves summing over all paths and is therefore robust to noise perturbations, 3) the distance takes into account all evidence relating *x* and *z*. $D_t(x, z)$ does not have to be computed explicitly. It can be computed using the eigenvectors and eigenvalues of *P*:

$$D_t^2(x, z) = \sum_{l \geq 1} \lambda_l^{2t} (\psi_l(x) - \psi_l(z))^2. \tag{8}$$

As previously mentioned, the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$ tend to 0 and have a modulus strictly less than 1. As a consequence, the above sum can be computed to a preset accuracy $\delta > 0$ with a finite number of terms: if we define as the number of elements retained to meet this accuracy. Then, up to relative precision δ, we have

$$D_t(x, z) = \left(\sum_{l \geq 1}^{s(\delta, t)} \lambda_l^{2t} (\psi_l(x) - \psi_l(z))^2 \right)^{\frac{1}{2}}. \tag{9}$$

We can therefore introduce a family of diffusion maps $\{\Psi_t\}_{t \in \mathbb{N}}$ given by

$$\psi_t : x \to \begin{pmatrix} \lambda_1^t \psi_1(x) \\ \lambda_2^t \psi_2(x) \\ \vdots \\ \lambda_{s(\delta,t)}^t \psi_{s(\delta,t)}(x) \end{pmatrix} \tag{10}$$

Each component of $\Psi_t(x)$ is termed diffusion coordinate. The map Ψ_t: $\Omega \to R^{s(\delta,t)}$ embeds the data set into a Euclidean space of $s(\delta, t)$ dimensions. This method constitutes a universal and data driven way to represent a graph, or any generic data set, as a cloud of points in a Euclidean space. Moreover, $s(\delta, t)$ depends on the properties of the random walk and not on the number of features of the original representation.

III. Kernel Functions

The kernel constitutes our prior definition of the local geometry of Ω, and since a given kernel will capture a specific feature of the data set, its choice should be guided by the application that one has in mind. Below is the list of kernels used here:

- Laplacian Kernel: $k(x,y) = \exp\left(-\left\|x-y\right\| - \mu \middle/ b\right) / 2b$,
- Gaussian Kernel: $k(x,y) = \exp(-\left\|x-y\right\|^2 / 2\sigma^2)$,
- Rayleigh Kernel: $k(x,y) = \dfrac{\left\|x-y\right\| \exp(-\left\|x-y\right\|^2 / 2\sigma^2)}{\sigma^2}$,
- Polynomial Kernel: $k(x,y) = (1+\langle x,y \rangle)^d$

where the Gaussian and Polynomial kernels are most familiar from support vector machines. The Laplacian and Rayleigh were introduced previously in [7].

IV. Experiments

A. Experimental Setup

The problem here is to analyze the effects on resultant diffusion maps of certain kernel functions for the classification of select databases. Each database is divided into ten groups that are as equal as possible, 10-fold cross validation. Nine groups are set aside for the training set and one group for the dedicated testing set. This procedure is continued until all groups have represented as a testing set. The average performance overall 10-folds is presented as the probability of classification (P_C), or sensitivity, and the probability of false alarm (P_{FA}), or specificity. This is done to demonstrate the trade-off between correctly classifying true cases versus incorrectly classifying false cases. Each kernel uses the same groups for each data set so that the possibility of poor individual performance due to the distribution of the draw is eliminated. In addition, each experiment is done ten times and the results are averaged over these runs.

A Linear Discriminant Analysis (LDA) classifier is used to evaluate the enhancement provided by the individual kernels to the diffusion map process. A LDA classifier assumes the classes have equal covariance matrices. In this case, the decision boundaries between classes is linear, and can in general be a hyperplane. The general form for LDA is

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma_k^{-1} \mu_k + \log \pi_k \tag{11}$$

and the decision rule is $G(x) = \text{argmax}_k \delta_k(x)$. Where the parameters are estimated from the training data as follows:

- $\pi_k = N_k / N$, where N_k is the number of class-k observations;
- $\mu_k = \sum_{g_i = k} x_i / N_k$;
- $\hat{\Sigma} = \sum_{k=1}^{K} \sum_{g_i = k} (x_i - \mu_k)(x_i - \mu_k)^T / (N - K)$.

For example, the LDA rule classifies to class 1 if

$$x^T \hat{\Sigma}^{-1} (\hat{\mu}_1 - \hat{\mu}_0) > \frac{1}{2} \mu_1^T \hat{\Sigma}^{-1} \mu_1 - \frac{1}{2} \mu_0^T \hat{\Sigma}^{-1} \mu_0 + \ldots + \log(N_0 / N) - \log(N_1 / N) \tag{12}$$

and class 0 otherwise.

The experimental variable values are listed below.

Experimental Variables
δ= 1e-7, α = 1, b = 2, μ = 1, σ^2 =3 , d = 3.

Where δ is the diffusion threshold, α is the diffusion probability distribution scaling, b is the Laplacian kernel scaling parameter, μ is the mean for the Laplacian kernel, σ^2 is the variance for the Gaussian kernel and the square of the mode for the Rayleigh kernel, and d is the polynomial kernel degree.

B. Data Sets

The experiment discussed above tests the kernels and their embeddings for classification enhancement on the resulting Diffusion Maps over eight publically available data sets [8]:

- Pima Indian: Pima Indian Diabetes
- Sonar1: Connectionist Bench Sonar
- WDBC: Wisconsin Diagnostic Breast Cancer
- WPBC: Wisconsin Prognostic Breast Cancer
- Clev. Heart: Heart Disease Data Set, Cleveland
- Wisc. BC: Wisconsin Breast Cancer Original
- Sonar2: Shallow Water Acoustic Toolset [9]
- Sonar3: Shallow Water Acoustic Toolset [9]

For each data set listed above, Table 1 below includes the number of samples, the class distribution, and the number of features, or attributes.

C. Results

The experimental results for the kernel effects on the resultant diffusion maps are shown below in Table 2 through Table 9. The tables are listed per database with each kernel given a column. The rows correspond to the original and reduced dimension pairs.

Table 1 shows that for the Pima Indian database the Polynomial and Gaussian kernels have a better P_C than the

Laplacian and Rayleigh kernels with a trade-off of a slightly worse P_{FA}. For the Sonar1 database, all of the kernels are fairly consistent with the Rayleigh kernel slightly outperforming, on average, the Laplacian kernel with an average P_C 72.6%. The Laplacian kernel outperforms the other three kernels on the WDBC database with an average P_C 98%, however as for an more acceptable P_{FA} the Rayleigh kernel offers a sound alternative with a decrease average P of 95.5%. This result differs from the WPBC database with the Rayleigh kernel resulting in an average P_C of 66% and all four kernels failing overall to capture the embedding appropriately.

Results for the Clev. Heart database show that the Rayleigh kernel captures the embedding with an average P_C of 77.3% and a slightly higher P_{FA} than the Gaussian kernel. For the Wisc. BC database the Gaussian kernel outperforms the other three with a P_C of 98.5% with a 0.4% increase in P_{FA} as compared to the next best Laplacian result. For the Sonar2 database the Rayleigh kernel outperforms the other three by a minimum of 13% for an average P_C of 95%. This demonstrates the superiority of this kernel to capture the embedding of this particular feature set. The performance on the Sonar3 database leaves much to be desired, however. With an average P_C of 79.4% the Rayleigh kernel demonstrates a marked improvement over the Gaussian with an average P_C of 51.7%, nevertheless the gain comes with an increased P_{FA} of 13.3%.

V. Conclusions and Future Work

As the experiments demonstrate, the choice of kernel effects the resultant diffusion map. Overall, the Laplacian and Rayleigh kernels outperformed the standard Polynomial and Gaussian kernels on all of these databases, with a few exceptions such as the Pima Indian and Wisc. BC datasets. It appears that the Laplacian and Rayleigh kernels perform best on the higher dimensional non-Gaussian datasets and the standard kernels work well with lower-dimensional data. Therefore, for enhanced target recognition capability and an acceptable P_{FA} the Rayleigh kernel appears the appropriate choice to best capture the embedding distribution to enhance the diffusion map process.

References

[1] R. Coifman; S. Lafon, "Diffusion Maps," Applied and Computational Harmonic Analysis, special issue on diffusion maps and wavelets, vol. 21, pp. 5-30, July 2006.

[2] R. Coifman; S. Lafon; A. Lee; M. Maggioni; B. Nadler; F. Warner; S. Zucker, "Geometric Diffusions as a Tool for Harmonics Analysis and Structure Definition of Data: Multiscale Methods," Proc. Nat"l Academy of Sciences, vol. 102, no. 21, pp. 7432-7437, May 2005.

[3] S. Roweis; L. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," Science, vol. 290, pp. 2323-2326, 2000.

[4] M. Belkin; P. Niyogi, "Laplacian Eigenmaps for dimensionality reduction and data representation," Neural Computation, v.15 n.6, p.1373-1396, June 2003.

[5] D. Donoho; C. Grimes, "Hessian Eigenmaps: New Locally Linear Embedding Techniques for High-Dimensional Data," Proc. Nat'l Academy of Sciences, vol. 100, no. 10, pp. 5591-5596, May 2003.

[6] Z. Zhang; H. Zha, "Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment," Technical Report CSE-02-019, Dept. of Computer Science and Eng., Pennsylvania State Univ., 2002.

[7] Isaacs, J.C.; Foo, S.Y.; Meyer-Baese, A., "Novel Kernels and Kernel PCA for Pattern Recognition," Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on , vol., no., pp.438-443, 20-23 June 2007

[8] A. Asuncion; D.J. Newman; (2007). UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science.

[9] G. Sammelmann; J. Christoff; J. Lathrop; "Synthetic Images of Proud Targets", Proc. IEEE/MTS OCEANS 2004, pp. 266-271.

Table 1. Experimental Data Sets

Data Set	# Samples	# Class 0	# Class 1	# Attributes
Pima Indian	768	268	500	8
Sonar1	208	97	111	60
WDBC	569	212	357	30
WPBC	198	151	47	33
Clev. Heart	303	164	139	13
Wisc. BC	699	458	241	9
Sonar2	22263	21154	1109	60
Sonar3	3562	3512	50	60

Table 2. Experimental Results For Pima Indian

	Kernel			
Dimension (Original,Final)	Gaussian	Laplacian	Rayleigh	Polynomial
(8,2)	PC: 0.558 FA: 0.5187	PC: 0.694 FA: 0.3881	PC: 0.672 FA: 0.347	PC: 0.748 FA: 0.4366
(8,3)	PC: 0.704 FA: 0.6567	PC: 0.704 FA: 0.3918	PC: 0.672 FA: 0.347	PC: 0.714 FA: 0.4067
(8,4)	PC: 0.702 FA: 0.4813	PC: 0.678 FA: 0.3806	PC: 0.666 FA: 0.3507	PC: 0.716 FA: 0.4067
(8,5)	PC: 0.724 FA: 0.4739	PC: 0.684 FA: 0.3246	PC: 0.676 FA: 0.3545	PC: 0.716 FA: 0.403

(8,6)	PC: 0.74 FA: 0.4515	PC: 0.692 FA: 0.291	PC: 0.674 FA: 0.3619	PC: 0.706 FA: 0.4216
(8,7)	PC: 0.744 FA: 0.4627	PC: 0.692 FA: 0.2873	PC: 0.7 FA: 0.306	PC: 0.704 FA: 0.3582
(8,8)	PC: 0.738 FA: 0.4813	PC: 0.684 FA: 0.2836	PC: 0.692 FA: 0.2873	PC: 0.692 FA: 0.3358

TABLE 3. EXPERIMENTAL RESULTS FOR SONAR1

	Kernel			
Dimension (Original,Final)	Gaussian	Laplacian	Rayleigh	Polynomial
(60,2)	PC: 0.5676 FA: 0.2887	PC: 0.5856 FA: 0.3711	PC: 0.6396 FA: 0.4227	PC: 0.5225 FA: 0.3505
(60,3)	PC: 0.6126 FA: 0.2577	PC: 0.7658 FA: 0.2268	PC: 0.7387 FA: 0.2577	PC: 0.6577 FA: 0.2165
(60,4)	PC: 0.7117 FA: 0.268	PC: 0.7568 FA: 0.2474	PC: 0.7387 FA: 0.2887	PC: 0.7568 FA: 0.2577
(60,5)	PC: 0.7748 FA: 0.2784	PC: 0.7477 FA: 0.268	PC: 0.7027 FA: 0.299	PC: 0.7297 FA: 0.2577
(60,6)	PC: 0.7477 FA: 0.299	PC: 0.7297 FA: 0.2268	PC: 0.7748 FA: 0.2784	PC: 0.7297 FA: 0.2371
(60,7)	PC: 0.7477 FA: 0.3196	PC: 0.7477 FA: 0.2165	PC: 0.7568 FA: 0.268	PC: 0.7297 FA: 0.2268
(60,8)	PC: 0.7477 FA: 0.3299	PC: 0.7297 FA: 0.2371	PC: 0.7297 FA: 0.268	PC: 0.7477 FA: 0.2371

TABLE 4. EXPERIMENTAL RESULTS FOR WDBC

	Kernel			
Dimension (Original,Final)	Gaussian	Laplacian	Rayleigh	Polynomial
(30,2)	PC: 0.9468 FA: 0.1462	PC: 0.9748 FA: 0.1651	PC: 0.9496 FA: 0.09906	PC: 0.9356 FA: 0.184
(30,3)	PC: 0.958 FA: 0.1604	PC: 0.9692 FA: 0.1132	PC: 0.958 FA: 0.08491	PC: 0.9888 FA: 0.2028
(30,4)	PC: 0.9468 FA: 0.1698	PC: 0.9804 FA: 0.1038	PC: 0.9608 FA: 0.08962	PC: 0.972 FA: 0.1321
(30,5)	PC: 0.958 FA: 0.1368	PC: 0.9832 FA: 0.1179	PC: 0.958 FA: 0.08962	PC: 0.9776 FA: 0.1179
(30,6)	PC: 0.9524 FA: 0.09906	PC: 0.9832 FA: 0.1226	PC: 0.9496 FA: 0.09434	PC: 0.9748 FA: 0.1274
(30,7)	PC: 0.9496 FA: 0.1085	PC: 0.9888 FA: 0.09434	PC: 0.944 FA: 0.08019	PC: 0.9748 FA: 0.1226
(30,8)	PC: 0.9552 FA: 0.09434	PC: 0.9804 FA: 0.09434	PC: 0.9636 FA: 0.08019	PC: 0.9776 FA: 0.1368

TABLE 5. EXPERIMENTAL RESULTS FOR WPBC

	Kernel			
Dimension (Original,Final)	Gaussian	Laplacian	Rayleigh	Polynomial
(33,2)	PC: 0.5532 FA: 0.3311	PC: 0.5532 FA: 0.3311	PC: 0.6596 FA: 0.3974	PC: 0.5745 FA: 0.351
(33,3)	PC: 0.5745 FA: 0.3046	PC: 0.5745 FA: 0.3377	PC: 0.7021 FA: 0.3576	PC: 0.5957 FA: 0.3775
(33,4)	PC: 0.5957 FA: 0.3444	PC: 0.5319 FA: 0.3709	PC: 0.6596 FA: 0.3709	PC: 0.617 FA: 0.3377

(33,5)	PC: 0.5957 FA: 0.3245	PC: 0.6383 FA: 0.351	PC: 0.6596 FA: 0.351	PC: 0.617 FA: 0.3444
(33,6)	PC: 0.6383 FA: 0.3311	PC: 0.617 FA: 0.3576	PC: 0.6596 FA: 0.3377	PC: 0.6383 FA: 0.3444
(33,7)	PC: 0.617 FA: 0.3311	PC: 0.6383 FA: 0.3576	PC: 0.6596 FA: 0.3377	PC: 0.617 FA: 0.351
(33,8)	PC: 0.6383 FA: 0.3113	PC: 0.617 FA: 0.3576	PC: 0.617 FA: 0.3311	PC: 0.617 FA: 0.3245

TABLE 6. EXPERIMENTAL RESULTS FOR CLEV. HEART

	Kernel			
Dimension (Original,Final)	Gaussian	Laplacian	Rayleigh	Polynomial
(13,2)	PC: 0.7266 FA: 0.122	PC: 0.7482 FA: 0.1951	PC: 0.7626 FA: 0.2622	PC: 0.741 FA: 0.189
(13,3)	PC: 0.7266 FA: 0.128	PC: 0.7482 FA: 0.189	PC: 0.7626 FA: 0.2256	PC: 0.7338 FA: 0.1768
(13,4)	PC: 0.7122 FA: 0.1402	PC: 0.7266 FA: 0.1768	PC: 0.7626 FA: 0.2134	PC: 0.7194 FA: 0.1707
(13,5)	PC: 0.7194 FA: 0.1463	PC: 0.7338 FA: 0.1646	PC: 0.7626 FA: 0.1768	PC: 0.7122 FA: 0.1646
(13,6)	PC: 0.7266 FA: 0.1341	PC: 0.7554 FA: 0.1585	PC: 0.7914 FA: 0.1768	PC: 0.7626 FA: 0.1524
(13,7)	PC: 0.7554 FA: 0.1341	PC: 0.7986 FA: 0.1646	PC: 0.7842 FA: 0.1768	PC: 0.7554 FA: 0.1463
(13,8)	PC: 0.741 FA: 0.1463	PC: 0.7986 FA: 0.1707	PC: 0.7842 FA: 0.128	PC: 0.7554 FA: 0.1463

TABLE 7. EXPERIMENTAL RESULTS FOR WISC. BC

	Kernel			
Dimension (Original,Final)	Gaussian	Laplacian	Rayleigh	Polynomial
(9,2)	PC: 0.9793 FA: 0.04367	PC: 0.971 FA: 0.03493	PC: 0.9585 FA: 0.02838	PC: 0.9378 FA: 0.02838
(9,3)	PC: 0.9793 FA: 0.03712	PC: 0.9668 FA: 0.03493	PC: 0.9585 FA: 0.02838	PC: 0.9544 FA: 0.03275
(9,4)	PC: 0.9876 FA: 0.03712	PC: 0.971 FA: 0.03057	PC: 0.9585 FA: 0.02838	PC: 0.9668 FA: 0.03275
(9,5)	PC: 0.9876 FA: 0.03712	PC: 0.9668 FA: 0.03493	PC: 0.9585 FA: 0.0262	PC: 0.9627 FA: 0.03275
(9,6)	PC: 0.9876 FA: 0.03493	PC: 0.971 FA: 0.03493	PC: 0.9585 FA: 0.02838	PC: 0.9668 FA: 0.03275
(9,7)	PC: 0.9876 FA: 0.03493	PC: 0.9668 FA: 0.03493	PC: 0.9627 FA: 0.02838	PC: 0.9668 FA: 0.03275
(9,8)	PC: 0.9876 FA: 0.03493	PC: 0.9668 FA: 0.03493	PC: 0.9627 FA: 0.03275	PC: 0.9585 FA: 0.03275

TABLE 8. EXPERIMENTAL RESULTS FOR SONAR2

	Kernel			
Dimension (Original,Final)	Gaussian	Laplacian	Rayleigh	Polynomial
(60,2)	PC: 0.4593 FA: 0.03258	PC: 0.7703 FA: 0.02744	PC: 0.9372 FA: 0.09654	PC: 0.5458 FA: 0.02247
(60,3)	PC: 0.7013 FA: 0.0269	PC: 0.7868 FA: 0.02901	PC: 0.9434 FA: 0.09768	PC: 0.7384 FA: 0.03004
(60,4)	PC: 0.7425 FA: 0.02761	PC: 0.8033 FA: 0.02685	PC: 0.9547 FA: 0.08947	PC: 0.7714 FA: 0.03198

(60,5)	PC: 0.759 FA: 0.02825	PC: 0.7951 FA: 0.02156	PC: 0.9392 FA: 0.07472	PC: 0.8208 FA: 0.02863
(60,6)	PC: 0.7765 FA: 0.03128	PC: 0.7981 FA: 0.0208	PC: 0.8877 FA: 0.05316	PC: 0.8043 FA: 0.02744
(60,7)	PC: 0.7621 FA: 0.02955	PC: 0.8012 FA: 0.0208	PC: 0.898 FA: 0.05451	PC: 0.7683 FA: 0.02048
(60,8)	PC: 0.7673 FA: 0.02944	PC: 0.8023 FA: 0.02075	PC: 0.898 FA: 0.05446	PC: 0.7775 FA: 0.02064

TABLE 9. EXPERIMENTAL RESULTS FOR SONAR3

	Kernel			
Dimension (Original,Final)	Gaussian	Laplacian	Rayleigh	Polynomial
(60,2)	PC: 0.52 FA: 0.09937	PC: 0.72 FA: 0.1193	PC: 0.82 FA: 0.1532	PC: 0.56 FA: 0.0803
(60,3)	PC: 0.58 FA: 0.08628	PC: 0.68 FA: 0.1233	PC: 0.82 FA: 0.1532	PC: 0.54 FA: 0.0660
(60,4)	PC: 0.48 FA: 0.07489	PC: 0.68 FA: 0.1136	PC: 0.82 FA: 0.1509	PC: 0.46 FA: 0.0996
(60,5)	PC: 0.48 FA: 0.07432	PC: 0.68 FA: 0.1079	PC: 0.84 FA: 0.1498	PC: 0.58 FA: 0.0896
(60,6)	PC: 0.48 FA: 0.07346	PC: 0.6 FA: 0.1065	PC: 0.78 FA: 0.1102	PC: 0.5 FA: 0.07574
(60,7)	PC: 0.54 FA: 0.06748	PC: 0.54 FA: 0.09539	PC: 0.74 FA: 0.1096	PC: 0.46 FA: 0.0674
(60,8)	PC: 0.54 FA: 0.07318	PC: 0.54 FA: 0.09653	PC: 0.74 FA: 0.1091	PC: 0.48 FA: 0.0620